08-18-04 AF IFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: Lyle Bate	§	Attorney Docket No.: 26530.4 (IDR-338)
Serial No.: 09/450,867	§	Customer No.: 27683
Filed: November 30, 1999	§	Group Art Unit: 2152
For: CACHING AND ACCESSING RIGHTS IN A DISTRIBUTED COMPUTING SYSTEM	§	Examiner: Willett, Stephan F.

BRIEF ON APPEAL

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

This Brief is submitted in connection with an appeal from the final rejection by the Examiner dated June 2, 2004, finally rejecting all of the pending claims 1-7, 10-18 and 23-38. Two additional copies of this Brief are also submitted.

REAL PARTY IN INTEREST

The real party in interest is NOVELL, INC., a Delaware corporation having a principal office and place of business at M/s PRV-F-331, 1800 South Novell Place, Provo, Utah, 84606-6194.

RELATED APPEALS AND INTERFERENCES

There are no related appeals and no related interferences regarding the above-identified patent application.

STATUS OF CLAIMS

Claims 1-7, 10-18 and 23-38 are pending, stand finally rejected, and are on appeal here. Claims 1-7, 10-18 and 23-38 are set forth in Appendix A attached hereto.

STATUS OF AMENDMENTS AFTER FINAL REJECTION

No claims were amended following the final rejection dated June 2, 2004.

SUMMARY OF THE INVENTION

The present invention, as now set forth in independent claim 1, relates to a method for caching and accessing access rights to at least one resource in a distributed computing system (Figs. 1 and 2). The method includes accessing, by a software agent, a directory service, wherein the agent is located on a deputization point coupled to the directory service, and wherein the directory service comprises the access rights of a software principal to a resource (Fig. 11; page 25, lines 8-11; Fig. 9; page 22, lines 15-17). The agent updates the access rights in an access control list cache, wherein the access control list cache is coupled to the deputization point and to the principal (Fig. 11; page 25, lines 11-13; Fig. 9; page 22, lines 17-18). A request is received at the access control list cache from the principal for the access rights stored in the access control list cache (Fig. 11; page 25, lines 13-15). The access rights are retrieved from the access control list cache, and forwarded to the principal (Fig. 11; page 25, lines 15-16; Fig. 9; page 22, lines 12-14). One or more of the principal's access rights are delegated to at least one software entity (Fig. 4; page 16, lines 4-7). The software entity accesses the resource using the delegated access rights without requiring intervention of the principal to authenticate access requests by the software entity, wherein tasks can be accomplished by the software entity without control by the principal. (Fig. 2; page 15, lines 13-17; Fig. 3; page 15, lines 24-26).

The present invention, as now set forth in independent claim 11, relates to a distributed computing system supporting access control caching (Figs. 1 and 2). The system comprises a plurality of computers, each having a memory and a processor (Fig. 1; page 11, lines 8-9 and lines 24-27; page 12, lines 1-8); a plurality of communication links connecting the plurality of computers (Fig. 1); a principal located on a first one of the computers (Figs. 1 and 2; page 12, lines 16-17); an agent located on a second one of the computers (Figs. 1 and 9; page 22, line 15); a resource located on a third one of the computers (Fig. 1); a first set of access rights located on a fourth one of the computers (Figs. 1 and 9; page 22, lines 12-14); a second set of access

rights located on a fifth one of the computers (Figs. 1 and 9; page 22, lines 12-14); means for accessing, by the agent, the first set of access rights of the principal to the resource (Fig. 9; page 22, lines 15-16); means for updating, by the agent, the first set of access rights to an access control list cache, wherein the access control list cache is located on a sixth one of the computers (Figs. 1 and 11; page 25, lines 11-13; Fig. 9; page 22, lines 17-18); means for receiving, at the access control list cache, a request from the principal for the first set of access rights (Fig. 11; page 25, lines 13-15; Fig. 9; page 22, lines 10-12); means for retrieving, by the access control list cache, the first set of access rights (Fig. 11; page 25, line 15; Fig. 9; page 22, lines 12-14); means for forwarding, to the principal, the first set of access rights (Fig. 11; page 25, line 16; Fig. 9; page 22, lines 12-14); and means for providing, to the principal, a deputization certificate adapted for enabling the principle to copy one or more of the principal's access rights to at least one software entity (Fig. 2; page 14, lines 25-27; page 15, lines 1-10).

The present invention, as now set forth in independent claim 15, relates to a computer storage medium having a configuration that represents data and instructions which will cause performance of method steps for caching and accessing access rights in a distributed computing system (Figs. 1 and 2). A software agent accesses a directory service, wherein the agent is located on a deputization point coupled to the directory service having the access rights of at least one principal to at least one resource (Fig. 11; page 25, lines 8-11; Fig. 9; page 22, lines 15-17). The agent updates the access rights to an access control list cache, wherein the access control list cache is coupled to the deputization point, and wherein the access control list cache is coupled to the principal (Fig. 11; page 25, lines 11-13; Fig. 9; page 22, lines 17-18). A request is received at the access control list cache from the principal for the access rights (Fig. 11; page 25, lines 13-15; Fig. 9; page 22, lines 10-12). The access rights are retrieved by the access control list cache, and forwarded to the principal (Fig. 11; page 25, lines 15-16; Fig. 9; page 22, lines 12-14). A deputization credential empowering the principal to deputize software entities is forwarded to the principal (Fig. 2; page 14, lines 25-27; page 15, lines 1-6). The principal deputizes at least one of the software entities, and

the software entity can exercise one or more of the principal's access rights due to the deputization (Fig. 2; page 15, lines 5-10).

The present invention, as now set forth in independent claim 23, relates to a method for controlling access within a computer system using deputization (Figs. 1 and 2). The method comprises receiving an access authorization request at a deputization point from a principal, wherein the access authorization request requests validation of the principal's identity (Fig. 2; page 12, lines 25-28; page 13, lines 1-3); determining whether to validate the principal based on the access authorization request (Fig. 2; page 13, lines 4-6); identifying one or more resource access permissions for the principal if the principal is validated, and the resource access permissions enable the principal to access one or more resources (Fig. 2; page 13, lines 10-15); and providing the principal with deputizing authority at the identified access authorization level, wherein the deputizing authority comprises a deputization credential that enables the principal to give at least one software entity within the computer system a level of resource access permission equal to or lesser than the principal's resource access permissions (Fig. 2; page 14, lines 25-27; page 15, lines 1-17).

The present invention, as now set forth in independent claim 29, relates to a computer-executable method for delegating permission from a software principal to a software deputy within a computer network to access at least one resource that is accessible to the principal (Figs. 1 and 2). A request is received from the principal for a deputy credential, wherein the request includes the principal's identity and at least one permission to be assigned to the deputy (Fig. 2; page 14, lines 3-13 and lines 19-21). The deputy credential is send to the principal, wherein the deputy credential enables the principal to assign the permission to the resource to the deputy (Fig. 2; page 14, lines 25-17; page 15, lines 1-6). A deputization request is received from the principal to assign the permission to the deputy (Fig. 4; page 15, lines 5-6; page 16, lines 4-7). Permission is assigned the to the deputy, and the deputy can independently access the resource using the assigned permission without being controlled by the principal (Fig. 2; page 15, lines 13-17 and lines 24-26).

ISSUES

Whether claims 1-7, 10-18 and 23-38 are unpatentable under 35 U.S.C. § 103(a) over U.S. Patent No. 6,178,510 to O'Connor et al. ("O'Connor") in view of U.S. Patent No. 6,157,953 to Chang et al. ("Chang").

GROUPING OF CLAIMS

As to the rejection of claims 1-7 and 10, it is Applicants' intention that solely for the purposes of this appeal, the rejected claims stand or fall together.

As to the rejection of claims 11-14, it is Applicants' intention that solely for the purposes of this appeal, the rejected claims stand or fall together.

As to the rejection of claims 15-18, it is Applicants' intention that solely for the purposes of this appeal, the rejected claims stand or fall together.

As to the rejection of claims 23-28, it is Applicants' intention that solely for the purposes of this appeal, the rejected claims stand or fall together.

As to the rejection of claims 29-38, it is Applicants' intention that solely for the purposes of this appeal, the rejected claims stand or fall together.

ARGUMENT

The Applicants' note that Applicants' intention that claims 1-7 and 10, 11-14, 15-18, 23-28 and 29-38 stand or fall separately is based on the following.

Claim 1 recites a limitation that is not recited in claims 11, 15, 23 or 29. More specifically, claim 1 recites "accessing the resource, by the software entity, using the delegated access rights without requiring intervention of the principal to authenticate access requests by the software entity, wherein tasks can be accomplished by the software entity without control by the principal." This limitation is not recited in claims 11, 15, 23 or 29.

Claims 11 recites a limitation that is not recited in claims 1, 15, 23 or 29. More specifically, claim 11 recites "a plurality of computers, each having a memory and a processor[.]" This limitation is not recited in claims 1, 15, 23 or 29.

Claims 15 recites a limitation that is not recited in claims 1, 11, 23 or 29. More specifically, claim 15 recites "deputizing, by the principal, at least one of the software

entities, wherein the software entity can exercise one or more of the principal's access rights due to the deputization."

Claims 23 recites a limitation that is not recited in claims 1, 11, 15 or 29. More specifically, claim 23 recites "receiving an access authorization request at a deputization point from a principal, wherein the access authorization request requests validation of the principal's identity[.]" This limitation is not recited in claims 1, 11, 15 or 29.

Claims 29 recites a limitation that is not recited in claims 1, 11, 15 or 23. More specifically, claim 29 recites "receiving a request from the principal for a deputy credential, wherein the request includes the principal's identity and at least one permission to be assigned to the deputy[.]" This limitation is not recited in claims 1, 11, 15 or 23.

Regarding the rejection of claim 1 under 35 U.S.C. § 112, Applicant respectfully submits that the use of "principal" to represent "software principal" is sufficiently clear in the context of the claim, as that is the only "principal" contained in claim 1. Therefore, Applicant respectfully submits that the rejection should be withdrawn.

Regarding the rejection of claim 34 under 35 U.S.C. § 112, Applicant respectfully submits that "the principal is terminated" is clear and is supported by page 15, lines 24-26 of the specification. Therefore, Applicant respectfully submits that the rejection should be withdrawn.

As discussed below, Applicant believes that the Examiner has improperly applied the combination of references to claims 1-7, 10-18 and 23-38. More specifically, it is Applicant's belief that the Examiner cannot factually support a prima facie case of obviousness with respect to claims 1-7, 10-18 and 23-38 because the references, even when combined, fail to teach or suggest the claimed subject matter.

INDEPENDENT CLAIMS

I. The Independent Claims Are Not Taught Or Suggested By O'Connor In View Of Chang

Claims 1, 11, 15, 23 and 29 stand rejected as being obvious under 35 U.S.C. §103(a) in light of O'Connor in view of Chang. It is well settled that, in order to reject a patent application for obviousness, the prior art references must teach or suggest all of

the claim limitations. *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). Moreover, all words in a claim must be considered in judging the patentability of that claim against the prior art. *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970). Accordingly, even if properly combinable, O'Connor in view of Chang must disclose all of the limitations of claims 1, 11, 15, 23 and 29, and all of the words in the claims must be considered when judging their patentability. When analyzing all of the claim limitations of claims 1, 11, 15, 23 and 29, it is clear that O'Connor and Chang fail to teach or suggest claims 1, 11, 15, 23 and 29 as required by MPEP § 2143.

A. Independent Claim 1 Is Not Taught Or Suggested By O'Connor In View Of Chang

Independent claim 1 is not taught by O'Connor in view of Chang for four independent reasons as set forth below.

First, claim 1 recites in part: "accessing, by a software agent, a directory service, wherein the agent is located on a deputization point coupled to the directory service, and wherein the directory service comprises the access rights of **a software principal** to a resource; . . . receiving, at the access control list cache, a request from **the principal** for the access rights stored in the access control list cache[.]" (emphasis added)

Instead of a **software principal**, the text of O'Connor cited by the Examiner describes **a user** desiring access to one or more of the hosts registers to receive access authorization to the desired host or hosts. (col. 7, lines 65-67) **The term "user" may include** multiple **persons** that meet the access criteria and that share a communication terminal. (col. 7, line 67 – col. 8, line 2) Applicants respectfully submit that since there exists a distinction between a **software principal** and a **person**, the art cited by the Examiner fails to include all the limitations of claim 1 as required by MPEP § 2143.

Second, claim 1 recites in part: "accessing, by a software agent, a directory service, **wherein the agent is located on a deputization point coupled to the directory service**, and **wherein the directory service comprises the access rights of a software principal to a resource**; updating, by the agent, the access rights in an

access control list cache, **wherein the access control list cache is coupled to the deputization point and to the principal**[.]" (emphasis added) However, the Examiner completely failed to recognize the above limitations, as he failed to cite any prior art that includes the above limitations. In fact, by requiring in-person identification (col. 8, lines 2-9), O'Connor teaches away from the limitation of **a directory service compris[ing] the access rights** as required by claim 1.

Third, claim 1 recites in part: "**updating, by the agent, the access rights in an access control list cache**, wherein the access control list cache is coupled to the deputization point and to the principal[.]" The examiner conceded in the Final Office Action (dated April 27, 2004) that O'Connor failed to teach updating rights. To overcome such deficiency, the Examiner relied on Chang. However, none of the cited text of Chang teaches the above limitations. For example, the Examiner cited col. 10, lines 31-34 of Chang: "[t]ypically an administrator will choose all the service hosts that contain services that were recently modified or added, and will enter all those service hosts at once from the browser host." However, that citation fails to include the above limitations. Accordingly, Applicants respectfully submit that the Examiner simply cited several pieces of irrelevant text that do not teach or suggest the above limitations in any plausible manner.

Fourth, claim 1 recites in part: "delegating one or more of the principal's access rights to at least one software entity; and **accessing the resource, by the software entity, using the delegated access rights without requiring intervention of the principal to authenticate access** requests by the software entity, wherein tasks can be accomplished by the software entity without control by the principal." (emphasis added) The cited text of O'Connor and Chang fails to teach or suggest the above elements. For example, O'Connor recites:

> At step 214, **the host acts on the user's instructions**. These instructions may include, but are not limited to, indicating what information the user wishes the host to retrieve, what services (e.g., gaming services) the user wants the host to provide, or what information the user wants the host to store or otherwise process. (col. 11, lines 18-24) (emphasis added)

Therefore, O'Connor simply discloses that the **host acts on the user's instructions**. It does not teach or suggest the delegation of rights to enable a software entity to **access a resource** using the delegated access rights **without requiring intervention or control by the principal**.

Since as stated above, claim 1 is allowable over the cited references based on any of the above four independent reasons, claim 1 should be allowed.

B. Independent Claim 11 is not taught or suggested by O'Connor in view of Chang

Independent claim 11 is not taught by O'Connor in view of Chang for three independent reasons as set forth below.

Claim 11 recites in part: "means for accessing, by the agent, the first set of access rights of the principal to the resource; **means for updating, by the agent, the first set of access rights to an access control list cache**, wherein the access control list cache is located on a sixth one of the computers; means for receiving, at the access control list cache, a request from the principal for the first set of access rights; means for retrieving, by the access control list cache, the first set of access rights; **means for forwarding, to the principal, the first set of access rights;** and **means for providing, to the principal, a deputization certificate adapted for enabling the principle to copy one or more of the principal's access rights to at least one software entity**." (emphasis added)

First, as discussed above with respect to claim 1, neither O'Connor nor Chang teaches or suggests "**means for updating, by the agent, the first set of access rights to an access control list cache**."

Second, the Examiner failed to cite any prior art that teaches or suggests "**means for forwarding, to the principal, the first set of access rights**[.]"

Third, the cited text of O'Connor fails to teach or suggest means for providing, to the principal, a deputization certificate adapted **for enabling the principle to copy one or more of the principal's access rights to at least one software entity**. For example, column 9, lines 8 to 11 of O'Connor recites the following: "to limit access to

portions of the host, **the gatekeeper may attach additional information** to the access request such as a code indicative of which portions of the host the user may access." (emphasis added)

Therefore, O'Connor allows **the gatekeeper** to attach additional information. In contrast, claim 11 requires "a deputization certificate adapted for **enabling the principle to copy** . . . access rights . . . ." (emphasis added) In addition, similar to the discussions above with respect to claim 11, O'Connor recites the access rights of **the user**, which differs from **a principal** that is located on a computer.

Since claim 11 is allowable over the cited references based on any of the above three independent reasons, claim 11 should be allowed.

C. Independent Claim 15 is not taught or suggested by O'Connor in view of Chang

Independent claim 15 is not taught by O'Connor in view of Chang for three independent reasons as set forth below.

Claim 15 recites in part: "**updating, by the agent, the access rights to an access control list cache**, wherein the access control list cache is coupled to the deputization point, and wherein the access control list cache is coupled to the principal; receiving, at the access control list cache, a request from the principal for the access rights; retrieving, by the access control list cache, the access rights; **forwarding, to the principal, the access rights**; **forwarding, to the principal, a deputization credential empowering the principal to deputize software entities**; and deputizing, by the principal, at least one of the software entities, wherein the software entity can exercise one or more of the principal's access rights due to the deputization." (emphasis added)

First, as discussed above with respect to claim 1, neither O'Connor nor Chang teaches or suggests "updating, by the agent, the access rights to an access control list cache[.]"

Second, the Examiner failed to cite any prior art that teaches or suggests "**forwarding, to the principal, the access rights**[.]"

Third, the cited text of O'Connor fails to teach or suggest "forwarding, to the principal, **a deputization credential empowering the principal to deputize software**

entities." Again, similar to the discussions above with respect to claim 11, column 9, lines 8 to 11 of O'Connor recites the following: "to limit access to portions of the host, **the gatekeeper may attach** additional information to the access request such as a code indicative of which portions of the host **the user** may access." (emphasis added)

Therefore, O'Connor allows **the gatekeeper** to attach additional information. In contrast, claim 15 requires "a deputization credential **empowering the principal to deputize** software entities[.]" (emphasis added)

Since claim 15 is allowable over the cited references based on any of the above three independent reasons, claim 15 should be allowed.

D. Independent Claim 23 is not taught or suggested by O'Connor in view of Chang

Independent claim 23 is not taught by O'Connor in view of Chang for the following reason as set forth below.

Claim 23 recites in part: "**providing the principal with deputizing authority** at the identified access authorization level, wherein **the deputizing authority comprises a deputization credential that enables the principal to give at least one software entity within the computer system a level of resource access permission** equal to or lesser than the principal's resource access permissions." (emphasis added)

The cited text of O'Connor fails to teach or suggest the above limitations. For example, column 9, lines 8 to 11 of O'Connor recites the following: "to limit access to portions of the host, **the gatekeeper may attach additional information** to the access request such as a code indicative of which portions of the host the user may access." (emphasis added)

Therefore, O'Connor allows **the gatekeeper** to attach additional information. In contrast, claim 23 requires "a deputization credential **that enables the principal to give** at least one software entity within the computer system a level of resource **access permission** . . . ." (emphasis added)

Therefore, claim 23 is allowable over the cited references, and should be allowed.

E. Independent Claim 29 is not taught or suggested by O'Connor in view of Chang

Independent claim 29 is not taught by O'Connor in view of Chang for three independent reasons as set forth below.

First, claim 29 recites in part: ""[a] computer-executable method for delegating permission from **a software principal** to a software deputy within a computer network to access at least one resource that is accessible to the principal [.]" (emphasis added)

Again, instead of a **software principal**, the cited text of O'Connor by the Examiner describes **a user** desiring access to one or more of the hosts registers to receive access authorization to the desired host or hosts. (col. 7, lines 65-67) **The term "user" may include** multiple **persons** that meet the access criteria and that share a communication terminal. (col. 7, line 67 – col. 8, line 2) Applicants respectfully submit that since there exists a distinction between a **software principal** and a **person**, the art cited by the Examiner fails to include all the limitations of claim 29 as required by MPEP § 2143.

Second, claim 29 recites in part: "sending the deputy credential to the principal, wherein **the deputy credential enables the principal to assign the permission to the resource to the deputy**[.]" For example, column 9, lines 8 to 11 of O'Connor recites the following: "to limit access to portions of the host, **the gatekeeper may attach additional information** to the access request such as a code indicative of which portions of the host the user may access." (emphasis added)

Therefore, O'Connor allows **the gatekeeper** to attach additional information. In contrast, claim 29 requires that the deputy credential **enables the principal to assign the permission** to the resource to the deputy.

Third, claim 29 recites in part: "assigning the permission to the deputy, wherein the deputy can independently access the resource using the assigned permission **without being controlled by the principal**." (emphasis added) However, the cited text of O'Connor and Chang fails to teach or suggest the above elements. For example, the cited text of O'Connor recites the following:

> At step 214, **the host acts on the user's instructions**. These instructions may include, but are not limited to, indicating what information the user wishes the host to retrieve, what services (e.g., gaming services) the user wants the host to provide, or what information the user wants the host to store or otherwise process. (col. 11, lines 18-24) (emphasis added)

Therefore, O'Connor simply discloses that the **host acts on the user's instructions**. It does not teach or suggest that the deputy can independently access the resource using the assigned permission **without being controlled by the principal**.

Since claim 29 is allowable over the cited references based on any of the above three independent reasons, claim 29 should be allowed.

II. Combination of O'Connor and Chang is Improper

The Examiner has taken the position that "it would have been obvious to one of ordinary skill in the art to incorporate updates as taught in Chang into the rights levels described in the O'Connor patent because O'Connor operates with security levels and Chang suggests that optimization can be obtained when capabilities are added." (page 4 of the Final Office Action dated April 27, 2004).

It is respectfully submitted that the combination of O'Connor and Chang is improper.

First, according to MPEP § 2143.01, obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention when some teaching, suggestion, or motivation to do so is found either explicitly or implicitly in the references themselves or in the knowledge generally available to one of ordinary skill in the art. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990).

Further, it is clear that there must be evidence that a skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention,

would select the elements from the cited prior art references for combination in the manner claimed. It is also clear that a rejection cannot be predicated on the mere identification of individual components of claimed limitations. Rather, **particular findings** must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed. *Ecolochem Inc. v. Southern California Edison,* 56 USPQ2d 1065, 1076 (Fed. Cir. 2000) (emphasis added)

Here, the Examiner failed to present such particular findings. In addition, there is no teaching, suggestion, or motivation to support the combination of O'Connor and Chang.

Second, the best defence against hindsight-based obviousness analysis is the rigorous application of the requirement for a showing of a teaching or motivation to combine the prior art references. See *In re Dembiczak*, 50 USPQ2d, 1614, 1617 (Fed. Cir. 1999). "Combining prior art references without ***evidence*** of such a suggestion, teaching, or motivation simply takes the inventor's disclosure as a blueprint for piecing together the prior art to defeat patentability – the essence of hindsight." *Id.* (emphasis added)

It is respectfully submitted that the only way O'Connor and Chang could be pieced together to defeat patentability is indeed to use Applicants' disclosure as a blueprint. Therefore, the combination of references is improper.

Third, *In re Dembiczak*, 50 USPQ 2d 1614, 1616-17 (Fed. Cir. 1999), the Federal Circuit held:

> Our analysis begins in the text of section 103 quoted above, with the phrase "at the time the invention was made." In this case, the Board fell into the hindsight trap. . . . The range of sources available, however, ***does not diminish the requirement for actual evidence. That is, the showing must be clear and particular.***

It is respectfully submitted that the Examiner failed to present clear and particular evidence in this case. A perfunctory assertion that "it would have been obvious to one of ordinary skill in the art to incorporate updates as taught in Chang into the rights levels described in the O'Connor patent because O'Connor operates with security levels and Chang suggests that optimization can be obtained when capabilities are added" is not

the clear and particular "evidence" demanded by the Federal Circuit. Thus, the Examiner failed to make a prima facie case for obviousness.

Accordingly, Applicants respectfully submit that the combination of O'Connor and Chang is improper.

Therefore, for this reason alone, claims 1, 11, 15, 23 and 29 are allowable over the cited references.

DEPENDENT CLAIMS

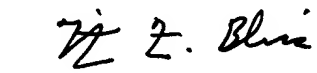
Dependent claims 2-7, 10, 12-14, 16-18, 24-28 and 30-38 depend from and further limit respective independent claims 1, 11, 15, 23 and 29, and should also be allowed.

III. Conclusion

Accordingly, it is respectfully submitted that neither O'Connor nor Chang teaches or suggests the subject matter of claims 1-7, 10-18 and 23-38. Moreover, it is respectfully submitted that it is improper to combine the references because there is no motivation or suggestion for such combination to achieve the Applicants' claimed elements.

For all of the foregoing reasons, it is respectfully submitted that claims 1-7, 10-18 and 23-38 be allowed. A prompt notice to that effect is earnestly solicited.

Respectfully submitted,



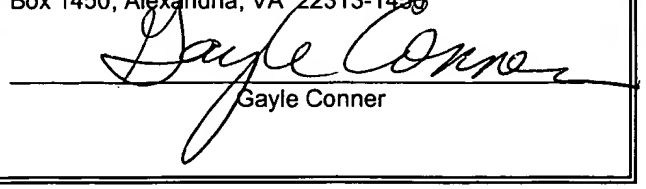
Timothy F. Bliss
Registration No. 50,925
Attorney for Applicants

Date: August 17, 2004
Haynes and Boone, LLP
901 Main Street, Suite 3100
Dallas, TX 75202-3789
(972) 739-8638

EXPRESS MAIL NO.:EV334578754US

DATE OF DEPOSIT: August 17, 2004

This paper and fee are being deposited with the U.S. Postal Service Express Mail Post Office to Addressee service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450



Gayle Conner

APPENDIX A

1. A method for caching and accessing access rights to at least one resource in a distributed computing system, the method comprising:

accessing, by a software agent, a directory service, wherein the agent is located on a deputization point coupled to the directory service, and wherein the directory service comprises the access rights of a software principal to a resource;

updating, by the agent, the access rights in an access control list cache, wherein the access control list cache is coupled to the deputization point and to the principal;

receiving, at the access control list cache, a request from the principal for the access rights stored in the access control list cache;

retrieving, from the access control list cache, the access rights;

forwarding, to the principal, the access rights;

delegating one or more of the principal's access rights to at least one software entity; and

accessing the resource, by the software entity, using the delegated access rights without requiring intervention of the principal to authenticate access requests by the software entity, wherein tasks can be accomplished by the software entity without control by the principal.

2. The method of claim 1, wherein the access control list cache is comprised of a first table comprising the principal that has access to the resource.

3. The method of claim 1, wherein the access control list cache is comprised of a second table comprising the access rights of the principal to the resource.

4. The method of claim 1, wherein the access control list cache is comprised of a third table comprising a cached access to the resource object.

5. The method of claim 2 further comprising invoking, by the directory service, a resource manager, if the first table does not contain the principal that has access to the resource, wherein the resource manager is coupled to the directory

service and comprises access information and access rights of the principal to the resource.

6. The method of claim 5 further comprising mapping, by the resource manager, an access control of the access rights in the resource manager to an access control of the rights in the directory service.

7. The method of claim 6 further comprising updating, by the resource manager, the mapped access control of the access rights to the access control list cache.

8 and 9 (Cancelled).

10. The method of claim 1, further comprising at least one of the following actions from the group consisting of:

asynchronously updating, by the agent to the access control list cache, the access rights, when the access rights are added to the directory service;

asynchronously updating, by the agent to the access control list cache, the access rights, when the access rights are removed from the directory service;

asynchronously updating, by the agent to the access control list cache, the access rights, when the request from the principal is received;

synchronously updating, by the agent to the access control list cache, the access rights, when the access rights are added to the directory service;

synchronously updating, by the agent to the access control list cache, the access rights, when the access rights are removed from the directory service;

synchronously updating, by the agent to the access control list cache, the access rights, when the request from the principal is received;

updating, at a scheduled time, the access rights by the agent to the access control list cache; and

updating, after a time to live has expired, the access rights by the agent to the access control list cache.

11. A distributed computing system supporting access control caching, the system comprises:

a plurality of computers, each having a memory and a processor;

a plurality of communication links connecting the plurality of computers;

a principal located on a first one of the computers;

an agent located on a second one of the computers;

a resource located on a third one of the computers;

a first set of access rights located on a fourth one of the computers;

a second set of access rights located on a fifth one of the computers;

means for accessing, by the agent, the first set of access rights of the principal to the resource;

means for updating, by the agent, the first set of access rights to an access control list cache, wherein the access control list cache is located on a sixth one of the computers;

means for receiving, at the access control list cache, a request from the principal for the first set of access rights;

means for retrieving, by the access control list cache, the first set of access rights;

means for forwarding, to the principal, the first set of access rights; and

means for providing, to the principal, a deputization certificate adapted for enabling the principle to copy one or more of the principal's access rights to at least one software entity.

12. The system of claim 11 further comprises means for invoking the second set of access rights, if the first set of access rights is not located on the fourth one of the computers.

13. The system of claim 12 further comprises means for mapping an access control of the second set of access rights to an access control of the first set of access rights.

14. The system of claim 13 further comprises, means for updating the access control list cache with the mapped access control of the first set of access rights.

15. A computer storage medium having a configuration that represents data and instructions which will cause performance of method steps for caching and accessing access rights in a distributed computing system, the method comprising:

accessing, by a software agent, a directory service, wherein the agent is located on a deputization point coupled to the directory service having the access rights of at least one principal to at least one resource;

updating, by the agent, the access rights to an access control list cache, wherein the access control list cache is coupled to the deputization point, and wherein the access control list cache is coupled to the principal;

receiving, at the access control list cache, a request from the principal for the access rights;

retrieving, by the access control list cache, the access rights;

forwarding, to the principal, the access rights;

forwarding, to the principal, a deputization credential empowering the principal to deputize software entities; and

deputizing, by the principal, at least one of the software entities, wherein the software entity can exercise one or more of the principal's access rights due to the deputization.

16. The configured storage medium of claim 15 further comprising invoking, by the directory service, a resource manager, if the access control list cache does not contain one of the access rights, wherein the resource manager is coupled to the directory service, and wherein the resource manager comprises the one right.

17. The configured storage medium of claim 16 further comprising mapping, by the resource manager, an access control of the one right to an access control of the access rights.

18. The configured storage medium of claim 17 further comprising updating, by the resource manager, the mapped access control of the access rights to the access control list cache.

19-22 (Cancelled).

23. A method for controlling access within a computer system using deputization, the method comprising:

receiving an access authorization request at a deputization point from a principal, wherein the access authorization request requests validation of the principal's identity;

determining whether to validate the principal based on the access authorization request;

identifying one or more resource access permissions for the principal if the principal is validated, wherein the resource access permissions enable the principal to access one or more resources; and

providing the principal with deputizing authority at the identified access authorization level, wherein the deputizing authority comprises a deputization credential that enables the principal to give at least one software entity within the computer system a level of resource access permission equal to or lesser than the principal's resource access permissions.

24. The method of claim 23 wherein determining whether to validate the principal includes comparing information present in the access authorization request to a plurality of access rights contained in an access control list cache.

25. (Previously presented): The method of claim 24 further comprising:

invoking a resource manager if the access control list cache does not contain an access right associated with the access authorization request;

locating the access right associated with the access authorization request; and

mapping the access right into the plurality of access rights.

26. The method of claim 23 further comprising deputizing, by the principal, a first software entity, wherein the first software entity has a level of resource access permission equal to or lesser than the principal's resource access permissions.

27. The method of claim 26 wherein deputizing includes defining a lifespan of the deputization.

28. The method of claim 26 further comprising deputizing, by the first software entity, a second software entity, wherein the second software entity has a level of resource access permission equal to or lesser than the first software entity's level of resource access permission.

29. A computer-executable method for delegating permission from a software principal to a software deputy within a computer network to access at least one resource that is accessible to the principal, the method comprising:

receiving a request from the principal for a deputy credential, wherein the request includes the principal's identity and at least one permission to be assigned to the deputy;

sending the deputy credential to the principal, wherein the deputy credential enables the principal to assign the permission to the resource to the deputy;

receiving a deputization request from the principal to assign the permission to the deputy; and

assigning the permission to the deputy, wherein the deputy can independently access the resource using the assigned permission without being controlled by the principal.

30. The method of claim 29 further comprising imposing a lifespan on the assignment of the permission, wherein the assignment will expire at the end of the lifespan.

31. The method of claim 29 further comprising imposing a lifespan on the deputy, wherein the deputy will terminate at the end of the lifespan.

32. The method of claim 29 further comprising:

determining if a deputy identified in the deputization exists; and

creating the deputy if the deputy does not exist.

33. The method of claim 32 further comprising identifying a start time in the deputization request for assigning the permission to the deputy, wherein the permission is not assigned to the deputy until the start time.

34. The method of claim 33 wherein the principal is terminated in the computer network prior to the start time.

35. The method of claim 29 further comprising verifying that the principal is permitted to access the resource prior to sending the deputy credential to the principal.

36. The method of claim 29 wherein the deputy is in a namespace that is not accessible to the principal, and wherein the deputy can use the permission to access a resource in the namespace that is not accessible to the principal.

37. The method of claim 29 wherein the request from the principal for a deputy credential includes a plurality of permissions to be assigned to the deputy, and wherein the deputy credential sent to the principal permits the principal to assign only a portion of the plurality of permissions to the deputy.

38. The method of claim 29 further comprising

receiving a second request from the principal for a second deputy credential, wherein the request includes the principal's identity and at least a second permission to be assigned to the deputy;

sending the second deputy credential to the principal; and

assigning the second permission contained in the second deputy credential to the deputy, wherein the deputy includes permissions from both the deputy credential and the second deputy credential.